CLAIMS

1. A programming interface embodied on one or more computer readable media, comprising:

a first group of services related to generating graphical objects;

a second group of services related to formatting content; and

a third group of services related to creating components of the graphical objects.

2. A programming interface as recited in claim 1, wherein the first group of services, the second group of services and the third group of services share a common programming model.

3. A programming interface as recited in claim 1, wherein the first group of services, the second group of services and the third group of services utilize a common markup language.

4. A programming interface as recited in claim 1, wherein the first group of services, the second group of services and the third group of services share a common event system.

5. A programming interface as recited in claim 1, wherein the first group of services, the second group of services and the third group of services share a common property definition system.

6. A programming interface as recited in claim 1, wherein the first group of services, the second group of services and the third group of services share a common input paradigm.

7. A programming interface as recited in claim 1, wherein the first group of services, the second group of services and the third group of services share a common system for nesting elements associated with a particular group of services within elements associated with another group of services.

8. A programming interface as recited in claim 1, wherein the first group of services includes a service that determines an appearance of the graphical objects.

9. A programming interface as recited in claim 1, wherein the first group of services includes a service that determines a behavior of the graphical objects.

10. A programming interface as recited in claim 1, wherein the first group of services includes a service that determines an arrangement of the graphical objects.

11. A programming interface as recited in claim 1, wherein the first group of services includes a plurality of nested elements that define the graphical objects.

12. A programming interface as recited in claim 1, wherein the graphical objects are comprised of one or more elements defined by vector graphics.

13. A programming interface as recited in claim 1, wherein the first group of services can define window properties in a markup language without launching a new window.

14. A programming interface as recited in claim 1, wherein the first group of services generate a user interface containing a plurality of graphical objects.

15. A programming interface as recited in claim 1, wherein the second group of services arrange the graphical objects.

16. A software architecture comprising the programming interface as recited in claim 1.

17. An application program interface embodied on one or more computer readable media, comprising:

a first group of services related to generating graphical objects;

a second group of services related to formatting content; and

a third group of services related to creating components of the graphical objects, wherein the first group of services, the second group of services and the third group of services share a common programming model.

18. An application program interface as recited in claim 17, wherein the first group of services, the second group of services and the third group of services utilize a common markup language.

19. An application program interface as recited in claim 17, wherein the third group of services includes services to generate geometric shapes.

20. An application program interface as recited in claim 17, wherein the second group of services includes arranging a plurality of data elements.

21. An application program interface as recited in claim 17, wherein the first group of services includes:

a service that determines an appearance of a graphical object; and

a service that determines a behavior of the graphical object.

22. An application program interface as recited in claim 17, wherein the first group of services includes a service that defines window properties in a markup language without launching a new window.

23. A computer system including one or more microprocessors and one or more software programs, the one or more software programs utilizing a programming interface to request services from an operating system, the programming interface including separate commands to request services consisting of the following groups of services:

a first group of services for generating graphical objects; and

a second group of services for creating components of the graphical objects, wherein the first group of services and the second group of services share a common programming model.

24. A computer system as recited in claim 23, wherein the first group of services includes:

a service for defining an appearance of the graphical objects; and

a service for defining an arrangement of the graphical objects.

25. A computer system as recited in claim 23, wherein the second group of services includes services to generate a plurality of geometric shapes.

26. A method comprising:

calling one or more first functions to facilitate generating graphical objects; and

calling one or more second functions to facilitate formatting content, wherein the first functions and the second functions share a common programming model.

27. A method as recited in claim 26, further including calling one or more third functions to facilitate creating components of the graphical objects.

28. A method as recited in claim 26, further including calling one or more third functions to facilitate generating geometric shapes contained in the graphical objects.

29. A method as recited in claim 26, wherein the first functions facilitate:

defining window properties in a markup language without launching a new window; and

generating a user interface containing a plurality of graphical objects.

30. A system comprising:

means for exposing a first set of functions that enable generating graphical objects; and

means for exposing a second set of functions that enable creating components of the graphical objects, wherein the components of the graphical objects include a plurality of geometric shapes, and wherein the first set of functions and the second set of functions share a common programming model.

31. A system as recited in claim 30, wherein the second set of functions further enable arrangement of the geometric shapes on a page to be rendered.

32. A system as recited in claim 30, further comprising means for exposing a third set of functions that enable formatting content for display.

33. A system as recited in claim 30, wherein the first set of functions and the second set of functions utilize a common markup language.

34. A system as recited in claim 30, wherein the first set of functions and the second set of functions share a common event system and a common property definition system.